

TP n° 1 - Commandes Unix

Pour cette séance, il vous est demandé d'ouvrir une fenêtre `terminal` et de travailler exclusivement dans cette fenêtre.

Pour chaque séance de TP n° *i*, vous devez créer un répertoire appelé TP*i* et travailler dans ce répertoire.

Pour cette séance, à partir de votre répertoire personnel, créer le répertoire TP1 comme suit :

```
mkdir TP1                                (mkdir : make Directory)
cd TP1                                   (cd : change Directory)
```

Exercice 1. Quelques commandes pour commencer :

- affichez votre nom de login avec la commande `whoami`
- affichez votre répertoire courant avec la commande `pwd`

Dans ce qui suit, vous travaillerez dans le répertoire TP1.

1. Créer un fichier `premierTexte` contenant une ou deux phrases. Pour cette création, on demande d'utiliser l'éditeur `nano` (`nano premierTexte`)
utilisation de nano : le prompt affichant le nom de la machine disparaît ; à la place, on a une page blanche où on peut taper du texte. Le nom du fichier que nous éditons et le nom de l'éditeur sont affichés en haut. La liste des commandes disponibles est écrite en bas. Vous pouvez faire `Ctrl-O` pour enregistrer et `Ctrl-X` pour quitter.
2. Pour visualiser le contenu de `premierTexte`, on peut utiliser la commande `less` (`less premierTexte`) ou la commande `more` (`more premierTexte`). Testez-les toutes les deux.
3. Quelle est la taille du fichier `premierTexte` ? Utiliser la commande
`ls -l premierTexte`

NB : Unix est *case sensitive*, ce qui signifie qu'il est sensible à la casse des caractères : minuscules et majuscules sont différentes. Par conséquent, par exemple `premierTexte` et `PremierTexte` sont deux noms de fichiers différents.

Exercice 2. Les commandes `cp`, `ls`, `mv`

1. faire une copie du fichier `premierTexte` appelée `introduction`, en utilisant la commande `cp`
`cp premierTexte introduction` (cp : copy)
comparer leur taille en tapant la commande `ls -l` (sans arguments, `ls` donne la liste de tous les fichiers du répertoire).
On peut aussi taper la commande `ls -l premierTexte introduction`
2. renommer `introduction` en `double` en utilisant la commande `mv`
`mv introduction double` (mv : move)
examiner le résultat de la commande `ls -l`
3. quelle différence y a-t-il entre `mv` et `cp` ?

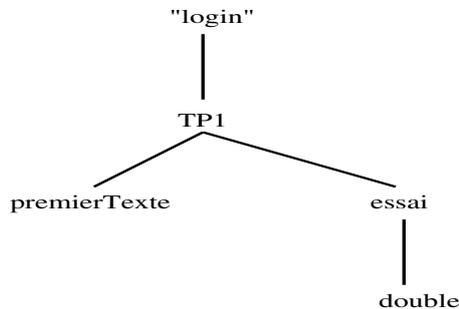
Exercice 3.

1. Dans le répertoire TP1, créer un répertoire appelé `essai`. Vérifiez que `essai` a bien été créé en utilisant la commande `ls`
2. déplacer le fichier `double` dans le répertoire `essai` avec la commande `mv`, puis déplacez-vous dans le répertoire `essai`.

```
mv double essai/  
cd essai
```

NB : `mv` sert à renommer des fichiers ainsi qu'à les déplacer. Le slash (/) n'est pas obligatoire, c'est un agrément pour identifier plus vite les répertoires dans une liste de fichiers.

3. À cette étape, nous avons construit l'arborescence suivante :



se placer de nouveau dans le répertoire TP1 avec la commande `cd`

```
cd ..
```

NB : `cd ..` permet de remonter au répertoire *père*, `cd ../..` remonte au répertoire *grand-père*, etc...

4. afficher la liste de ce que contient le répertoire `essai` en utilisant `ls`
5. faire une copie du fichier `premierTexte` dans le répertoire `essai` sous le nom de `copie`, et afficher de nouveau le contenu du répertoire `essai`

```
cp premierTexte essai/copie  
ls essai
```

Exercice 4. On reste dans le répertoire TP1

1. tenter de détruire le répertoire `essai` avec la commande `rm`. Que se passe-t-il ?

C'est la commande `rmdir` qui sert à supprimer un répertoire. Essayons avec la commande `rmdir essai`. De nouveau, c'est un échec. De cette manière, on ne peut pas détruire le répertoire avant de le vider. 2 solutions :

- a) soit on vide le répertoire `essai` avant de le détruire,

```
rm essai/*  
rmdir essai
```

- b) soit on le fait avec une seule commande en tapant la commande

```
rm -r essai
```

L'option `-r` est récursive. Elle supprime toute l'arborescence à partir du répertoire `essai`

Attention ! préférez l'utilisation de la commande `rm` avec l'option interactive, i.e. `rm -i`. Avec l'option `-i`, le système demande confirmation pour supprimer le(s) fichier(s).

Exercice 5. Les caractères magiques

Créer un répertoire `lesSujets` sous le répertoire TP1 et placez-vous dans ce répertoire.

1. En utilisant `nano` ou en effectuant des copies de fichiers, créer des fichiers supplémentaires dans le répertoire `tp1` que vous appellerez `subj1`, `subj2`, `subj3`, `subj12`, `subj7`, `subj21`, `subj4`, `subj30`, `subj32`, `objet201` et `objet`. Tester les commandes

```
ls s*           ls s?           ls subj?       ls subj[0-9]
ls subj[0-9]*   ls subj[0-9]?   ls [a-z][a-z]?[0-9]
ls [^a-zA-Z]
```

2. effacer en une seule commande les fichiers `subj3`, `subj30` et `subj32`
3. afficher en une seule commande les fichiers dont le nom contient un numéro compris entre 20 et 99

Exercice 6. Les droits d'accès

Créer dans le répertoire TP1 un fichier `tp1.txt` en utilisant `nano` et affichez ses droits d'accès. Utiliser la commande `chmod` pour modifier ces droits d'accès et leur donner les valeurs suivantes :

```
rwxr-xr-x  
rwxr-x---  
rw-r--r--
```

Exercice 7. À partir du répertoire TP1 :

1. Lister les répertoires `bin` et `dev`.
2. Lister de manière détaillée le contenu du répertoire `/etc`.
Les fichiers nommés `passwd`, `shadow`, `group` et `gpasswd` sont-ils présents dans `/etc` ?
3. Créer l'arborescence supplémentaire suivante dans TP1, où le fichier nommé `monFichier` est un fichier texte créé avec l'éditeur `nano` qui contiendra quelques lignes de votre cru et dans laquelle `monFichier.old` est la copie de `monFichier`.
4. Afficher les droits de `monFichier` et donner la permission au groupe de le modifier si nécessaire.

